ISSN 2541-1004

PENERAPAN RESAMPLING DAN ADABOOST UNTUK PENANGANAN MASALAH KETIDAKSEIMBANGAN KELAS BERBASIS NAÏVE BAYES PADA PREDIKSI CHURN PELANGGAN

Sri Mulyati[1], Yulianti[2], dan Aries Saifudin[3]
[1,2,3]Teknik Informatika, Universitas Pamulang
email: [1]city_oye@yahoo.com, [2]yulianti.saifudin@gmail.com, [3]aries.saifudin@gmail.com

ABSTRAK

Banyaknya operator seluler mendorong persaingan usaha yang sangat ketat. Kemudahan pelanggan untuk berpindah ke pesaing merupakan perhatian utama bagi bagian CRM (*Customer Relationship Management*), karena untuk mendapatkan pelanggan baru membutuhkan biaya yang jauh lebih mahal daripada mempertahankan pelanggan yang sudah ada. Untuk mengambil tindakan yang tepat dalam mempertahankan pelanggan harus mengetahui kecenderungan pelanggan apakah akan mengalami *churn* atau tidak. Prediksi kecenderungan pelanggan dilakukan dengan menggunakan model data mining. Pada penelitian ini akan diterapkan teknik *resampling* dan teknik *ensemble* AdaBoost untuk memperbaiki kinerja pengklasifikasi sedangkan untuk mengukur kinerja model digunakan software RapidMiner. Hasil penelitian menunjukkan bahwa model integrasi *random oversampling*, AdaBoost, dan Naïve Bayes memiliki kinerja yang lebih baik karena memiliki nilai AUC (*Area Under the ROC (Receiver Operating Characteristic) Curve*) yang lebih baik.

Kata Kunci: AdaBoot, Churn Pelanggan, Naïve Bayes, Prediksi, Resampling

1. PENDAHULUAN

Teknologi dan jumlah pengguna media telekomunikasi telah mengalami peningkatan secara signifikan. Jumlah pengguna telepon seluler diperkirakan mencapai lebih dari 7 milyar pada tahun 2015 (Sanou, 2015, p. 1) (Nistanto, 2014, p. 1), karena alat telekomunikasi dapat merangsang pertumbuhan ekonomi secara signifikan dan bahkan telah menjadi salah satu faktor keberhasilan pembangunan suatu bangsa. Dengan demikian telepon selular telah membawa masyarakat menuju kehidupan modern yang mengutamakan efisiensi dan kepraktisan.

Banyaknya operator seluler mendorong persaingan usaha yang sangat ketat. Pelanggan dapat memilih di antara beberapa penyedia layanan dan secara aktif menggunakan hak mereka beralih dari satu penyedia layanan ke yang lainnya. Terbukanya persaingan bebas diperusahaan jasa telekomunikasi merupakan salah satu tantangan serius yang harus dihadapi oleh industri telekomunikasi (Huang, Kechadi, & Buckley, 2012, p. 1414). Kemudahan pelanggan untuk berpindah ke pesaing merupakan perhatian utama bagi bagian CRM (*Customer Relationship Management*) (Jadhav & Pawar, 2011, p. 17), karena untuk mendapatkan pelanggan baru biayanya lebih mahal lima sampai enam kali lipat daripada mempertahankan pelanggan yang sudah ada (Verbeke, Dejaeger, Martens, Hur, & Baesens, 2012, p. 211). Hal ini juga telah menjadi isu penting dan merupakan salah satu tantangan utama perusahaan yang harus dihadapi di era global ini. Dalam pasar yang sangat kompetitif ini, pelanggan menuntut produk yang disesuaikan, dan layanan yang lebih baik dengan harga yang lebih murah, sementara penyedia layanan terus fokus pada akuisisi sebagai tujuan bisnis mereka.

Mengingat fakta bahwa industri telekomunikasi mengalami rata-rata tingkat *churn* tahunan 30-35 persen, dan biaya untuk merekrut pelanggan baru 5-10 kali lebih mahal daripada mempertahankan yang sudah ada, maka mempertahankan pelanggan menjadi lebih penting daripada mengakuisisi pelanggan (Lu, 2002, p. 1). *Churn* pelanggan mengacu pada hilangnya pelanggan secara periodik dalam suatu organisasi (Churi, Divekar, Dashpute, & Kamble, 2015, p. 225) (Yu, Guo, Guo, & Huang, 2010, p. 1425). *Churn* pelanggan merupakan penyebab kebocoran pendapatan terbesar dari perusahaan telekomunikasi (Jadhav & Pawar, 2011, p. 17). Untuk mempertahankan pelanggan yang sudah ada, organisasi harus meningkatkan layanan pelanggan, memperbaiki kualitas produk, dan harus dapat mengetahui lebih awal pelanggan mana yang memiliki kemungkinan akan meninggalkan organisasi.

Prediksi *churn* dapat digunakan untuk mengidentifikasi *churners* lebih awal sebelum mereka berpindah, dan dapat membantu departemen CRM (*Customer Relationship Management*) untuk mempertahankan mereka, sehingga potensi kerugian perusahaan dapat dihindari (Umayaparvathi & Iyakutti, 2012, p. 6). Prediksi *churn* pelanggan merupakan strategi bisnis yang penting bagi perusahaan. Dengan melakukan prediksi *churn* pelanggan, maka perusahaan dapat segera mengambil tindakan untuk mempertahankan pelanggan.

Untuk melakukan prediksi menggunakan teknik *data mining* diperlukan data-data masa lalu yang telah dikumpulkan. Data-data konsumen banyak tersedia di dalam database perusahaan, bagaimana menggunakannya untuk memprediksi *churn* pelanggan merupakan tantangan bagi para peneliti (Chen, Fan, & Sun, 2012, p. 461). Tetapi untuk mendapatkan dataset pelanggan yang sebenarnya merupakan masalah yang sulit bagi peneliti, karena dapat disalahgunakan (Umayaparvathi & Iyakutti, 2012, p. 6). Sehingga sebagian peneliti menggunakan dataset *churn* pelanggan yang telah disediakan untuk umum di internet.

Telah banyak algoritma klasifikasi yang telah diterapkan, seperti pengklasifikasi berdasarkan aturan (Ripper, PART), pendekatan pohon keputusan (C4.5, CART, *Alternating Decision Trees*), jaringan saraf (*Multilayer Perceptron, Radial Basis Function Network*), *nearest neighbor* (kNN), metode *ensemble* (*Random Forests, Logistic Model Tree*, Bagging, Boosting), dan metode statistik klasik (*logistic regression*, Naïve Bayes, *Bayesian Networks*) (Verbeke, Dejaeger, Martens, Hur, & Baesens, 2012, p. 212). Metode statistik klasik seperti logistic regression dan Naïve Bayes memberikan nilai yang baik, hasil yang kuat, dan mudah diterapkan (Verbeke, Dejaeger, Martens, Hur, & Baesens, 2012, p. 227), tetapi belum mencapai nilai yang sangat baik (*excellent*).

Data yang dikumpulkan untuk prediksi churn biasanya tidak seimbang, di mana kasus pelanggan *non-churn* jauh lebih banyak daripada kasus untuk pelanggan yang *churn* (Rodan, Fayyoumi, Faris, Alsakran, & Al-Kadi, 2015, p. 2). Jika dilihat pada dataset yang tersedia, persentase data *churn* dan persentase data yang bukan *churn* terlihat tidak seimbang, karena persentase data churn hanya sekitar 14,49%. Klasifikasi data dengan pembagian kelas yang tidak seimbang dapat menimbulkan penurunan kinerja yang signifikan yang dicapai oleh algoritma belajar (*learning algorithm*) pengklasifikasi standar, yang mengasumsikan distribusi kelas yang relatif seimbang dan biaya kesalahan klasifikasi yang sama (Sun, Mohamed, Wong, & Wang, 2007, p. 3358). Ketepatan parameter tidak dapat digunakan untuk mengevaluasi kinerja dataset yang tidak seimbang (Catal, 2012, p. 195).

Ada 2 pendekatan untuk menangani ketidakseimbangan kelas, yaitu pendekatan level data, dan pendekatan level algoritma (Peng & Yao, 2010, p. 111). Pendekatan pada level data mencakup berbagai teknik *resampling*, memanipulasi data latih untuk memperbaiki kecondongan distribusi kelas, seperti *random oversampling* dan *random undersampling* (Chawla, Bowyer, Hall, & Kegelmeyer, 2002, p. 328). Pada pendekatan algoritma, tujuannya adalah menyesuaikan operasi algoritma yang ada untuk membuat pengklasifikasi (*classifier*) agar lebih konduktif terhadap klasifikasi kelas minoritas (Zhang, Liu, Gong, & Jin, 2011, p. 2205).

Pendekatan level algoritma dapat menggunakan teknik menggabungkan atau memasangkan (*ensemble*) metode. Ada dua algoritma *ensemble-learning* paling populer, yaitu *boosting* dan *bagging* (Yap, et al., 2014, p. 14). Algoritma *boosting* telah dilaporkan sebagai meta-teknik untuk mengatasi masalah ketidakseimbangan kelas (*class imbalance*) (Sun, Mohamed, Wong, & Wang, 2007, p. 3360). AdaBoost diterapkan pada Naïve Bayes dapat meningkatkan kinerja sebesar 33,33% dan menghasilkan hasil yang akurat dengan mengurangi nilai kesalahan klasifikasi dengan meningkatkan iterasi (Korada, Kumar, & Deekshitulu, 2012, p. 73).

Berdasarkan uraian di atas, maka pada penelitian ini akan diterapkan teknik *resampling* dan AdaBoost. Teknik *resampling* yang digunakan adalah *random oversampling* dan *random undersampling*. Sedangkan algoritma dasar yang digunakan untuk pengklasifikasi adalah Naïve Bayes. Metode yang diusulkan diharapkan dapat mengurangi pengaruh ketidakseimbangan kelas terhadap kinerja pengklasifikasi, sehingga kinerja pengklasifikasi Naïve Bayes dapat meningkat dalam memprediksi *churn* pelanggan.

2. PENELITIAN TERKAIT

Pada penelitian yang dilakukan oleh Keramati, dkk (Keramati, et al., 2014) dinyatakan bahwa untuk bertahan dalam bisnis telekomunikasi harus dapat membedakan antara pelanggan yang memiliki kemungkinan untuk berpindah ke pesaing, dan pelanggan yang enggan untuk

berpindah. Oleh karena itu, prediksi kecenderungan pelanggan mengalami *churn* telah menjadi isu penting dalam bisnis telekomunikasi. Dalam bisnis yang kompetitif, prediktor pelanggan yang handal dianggap tidak ternilai harganya. Pada penelitian ini menerapkan teknik klasifikasi *data mining* termasuk Decision Tree (DT), Artificial Neural Networks (ANN), k-Nearest Neighbors (k-NN), dan Support Vector Machine (SVM), kemudian membandingkan kinerjanya. Dataset yang digunakan diperoleh dari perusahaan telekomunikasi di Iran. Selain itu, pada penelitian ini juga mengusulkan metode *hybrid* yang ditujukan untuk membuat perbaikan nilai ukuran evaluasi. Hasil penelitian menunjukkan bahwa nilai *precision* dan *recall* dapat mencapai nilai yang sangat baik.

Pada penelitian yang dilakukan oleh Rodan, Fayyoumi, Faris, Alsakran, dan Al-Kadi (Rodan, Fayyoumi, Faris, Alsakran, & Al-Kadi, 2015) dinyatakan bahwa saat ini perusahaan telekomunikasi telah menaruh banyak perhatian terhadap masalah identifikasi perilaku *churn* pelanggan. Dalam bisnis hal ini telah diketahui bahwa menarik pelanggan baru jauh lebih mahal daripada mempertahankan yang sudah ada. Oleh karena itu, menerapkan model prediksi *churn* pelanggan yang akurat secara efektif dapat membantu dalam mempertahankan pelanggan dan meningkatkan keuntungan. Pada penelitian ini digunakan ensemble *Mutilayer Perceptrons* (MLP) dengan pelatihan yang diperoleh menggunakan *Negative Correlation Learning* (NCL) untuk memprediksi *churn* pelanggan di dalam perusahaan telekomunikasi. Hasil penelitian menegaskan bahwa NCL berbasis ensemble MLP dapat mencapai kinerja secara umum lebih baik dibandingkan dengan ensemble MLP tanpa NCL (*flat ensemble*) dan teknik *data mining* umum yang digunakan untuk menganalisa *churn* pelanggan.

3. METODE PENELITIAN

Pada penelitian ini menggunakan teknik *resampling* dan teknik *ensemble* AdaBoost untuk memperbaiki kinerja pengklasifikasi Naïve Bayes. Untuk mengukur/menganalisa kinerja model yang diusulkan, digunakan aplikasi RapidMiner. Dataset churn pelanggan industri telekomunikasi yang digunakan diperoleh dari https://bigml.com/dashboard/source/55c69eca200d5a25a0005180.

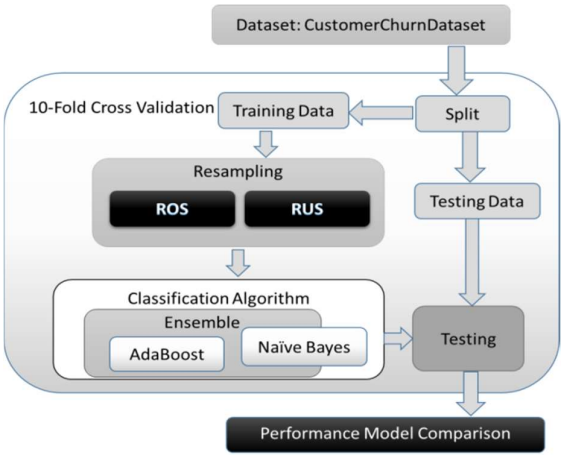
Spesifikasi dataset churn pelanggan industri telekomunikasi yang diperoleh tunjukkan pada Tabel 1.

Tabel 1 Spesifikasi dan Atribut Dataset Churn Pelanggan

No.	Atribut	Keterangan	Contoh
1	Phone	No. Telepon	382-4657
2	State	Kode 51 negara bagian (*district*) Columbia	KS
3	Account Length	Lama akun aktif	128
4	Area Code	Kode area	415
5	Int'l Plan	Rencana/mengaktifkan panggilan internasional; 1=Ya 0=Tidak	0
6	VMail Plan	Rencana/mengaktifkan pesan suara (voice mail); 1=Ya 0=Tidak	1
7	VMail Message	Pesan suara	25
8	Day Mins	Lama panggilan siang hari (menit)	265,1
9	Day Calls	Jumlah panggilan siang hari	110
10	Day Charge	Biaya panggilan siang hari	45,07
11	Eve Mins	Lama panggilan sore hari (menit)	197,4
12	Eve Calls	Jumlah panggilan sore hari	99
13	Eve Charge	Biaya panggilan sore hari	16,78
14	Night Mins	Lama panggilan malam hari (menit)	244,7
15	Night Calls	Jumlah panggilan malam hari	91
16	Night Charge	Biaya panggilan malam hari	11,01
17	Intl Mins	Lama panggilan Internasional (menit)	10
18	Intl Calls	Jumlah panggilan internasional	3
19	Intl Charge	Biaya panggilan internaional	2,7
20	CustServ Calls	Jumlah panggilan ke layanan pelanggan (customer service)	1

No.	Atribut	Keterangan	Contoh
21	Churn	Status churn (0=tidak/1=ya)	0

Kerangka penelitian ditunjukkan pada Gambar 1. Dataset churn pelanggan yang menjadi masukan dibagi menjadi 10 sesuai nilai validasi (*10-Fold Cross Validation*), satu bagian (1/10) digunakan sebagai data uji (*testing*), sisanya digunakan sebagai data latih (*training*). Kemudian data latih digunakan untuk melatih model yang diusulkan. Model yang sudah dilatih selanjutnya diuji dengan data uji untuk melakukan validasi. Hasil validasi digunakan untuk mengukur kinerja masing-masing model, dan dilakukan perbandingan kinerja untuk mencari model yang memiliki kinerja terbaik.



Gambar 1 Kerangka Penelitian

K-fold cross validation adalah teknik umum untuk memperkirakan kinerja pengklasifikasi. *K-fold cross validation* dilakukan dengan menggunakan kembali dataset yang sama, sehingga menghasilkan k perpecahan dari kumpulan data menjadi *non-overlapping* dengan proporsi pelatihan (k-1)/k dan 1/k untuk pengujian (Korb & Nicholson, 2011, p. 213).

Misalnya diberikan sekumpulan m data latih, dan menjalankan *k-fold cross validation* mengikuti proses berikut ini:

1. Mengatur contoh pelatihan dalam urutan acak.
2. Membagi data latih menjadi k lipatan. (k potongan masing-masing sekitar m/k data latih).
3. For i = 1; : : : ; k:
   a. Latih pengklasifikasi menggunakan semua data latih yang bukan milik potongan ke-i.
   b. Uji pengklasifikasi dari semua data latih menggunakan potongan ke-i.
   c. Hitung n_i, jumlah dari data latih dalam potongan ke-i yang diklasifikasikan salah.
4. Kembalikan perkiraan kesalahan pengklasifikasi sesuai rumus:

$$E = \frac{\sum_{i=1}^{k} n_i}{m} \qquad (1)$$

Untuk mendapatkan perkiraan yang akurat dari pengklasifikasi, *k-fold cross validation* dijalankan beberapa kali, masing-masing dengan pengaturan acak yang berbeda pada langkah 1. Misalnya E_1; :::, E_t menjadi perkiraan akurasi yang diperoleh t proses. Didefinisikan:

$$e = \frac{\sum_{j=1}^{t} E_j}{t} \qquad (2)$$

$$V = \frac{\sum_{j=1}^{t} (E_j - e)^2}{t-1} \qquad (3)$$

$$\sigma = \sqrt{V} \qquad (4)$$

Perkiraan kinerja algoritma adalah kesalahan e dengan standar deviasi σ. Berdasarkan pengujian yang luas dengan berbagai dataset yang berbeda, dan teknik belajar yang berbeda, telah menunjukkan bahwa 10 merupakan jumlah lipatan yang tepat (Witten, Frank, & Hall, 2011, p. 153). Dengan *10-fold cross validation*, pengukuran hasil dapat lebih akurat karena data yang ada dibagi ke dalam 10 data dengan jumlah yang sama, kemudian satu persatu diambil untuk pengujian, dan 9 bagian lainnya digunakan untuk pelatihan. Dengan *cross-validation* akurasi dari hasil pengukuran data akan lebih terjamin karena mengurangi kemungkinan data yang tidak konsisten dalam tahap prediksi.

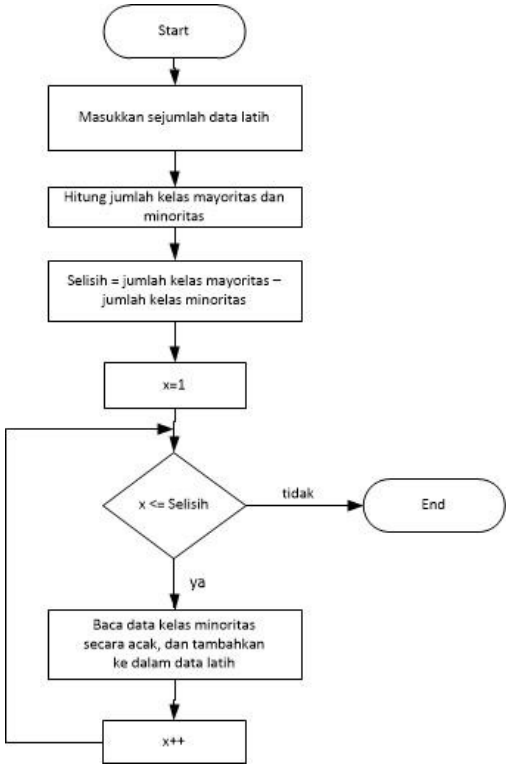
Algoritma ROS (Random Over-Sampling)

ROS bekerja dengan memilih data dari kelas minoritas secara acak dan menambahkannya ke data latih sehingga jumlah data kelas minoritas sama dengan jumlah data kelas mayoritas.

Algoritma ROS (*Random Over-Sampling*)

1. Masukkan sejumlah data latih
2. Hitung jumlah kelas mayoritas dan minoritas
3. Hitung Selisih antara jumlah kelas mayoritas dengan jumlah kelas minoritas
4. Baca data kelas minoritas secara acak dan tambahkan ke dalam data latih selama

masih terjadi selisih (sampai kelas minoritas sama dengan mayoritas)



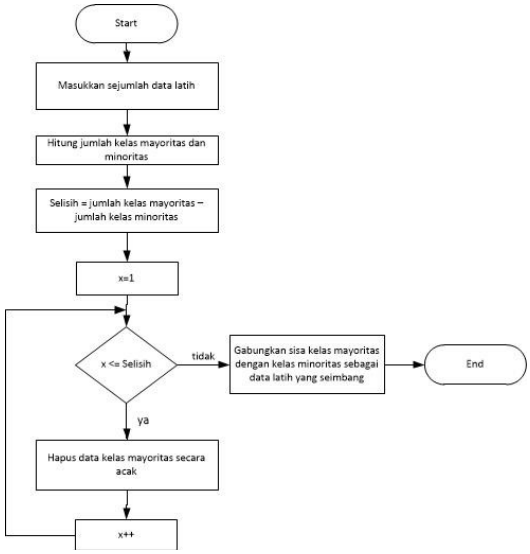
Gambar 2 Flowchart Algoritma ROS (*Random Over-Sampling*)

Algoritma RUS (Random Under-Sampling)

Sedangkan RUS bekerja dengan memilih data kelas mayoritas secara acak dan mengeluarkannya dari data latih hingga data kelas mayoritas sama dengan data kelas minoritas.

Algoritma RUS (*Random Under-Sampling*)

1. Masukkan sejumlah data latih
2. Hitung jumlah kelas mayoritas dan minoritas
3. Hitung Selisih antara jumlah kelas mayoritas dengan jumlah kelas minoritas
4. Hapus data kelas mayoritas selama masih terjadi selisih (sampai kelas minoritas sama dengan mayoritas)
5. Gabungkan sisa kelas mayoritas dengan kelas minoritas sebagai data latih yang seimbang



Gambar 3 Flowchart Algoritma RUS (*Random Under-Sampling*)

Tahap selanjutnya adalah melakukan klasifikasi untuk memprediksi keluaran variable/kelas. Algoritma yang digunakan adalah AdaBoost dan Naïve Bayes.

AdaBoost

AdaBoost (*Adaptive Boosting*) merupakan algoritma *machine learning* yang dirumuskan oleh Yoav Freund dan Robert Schapire (Afza, Farid, & Rahman, 2011, p. 105) (Harrington, 2012, p. 132). Algoritama AdaBoost merupakan algoritma yang membangun pengklasifikasi kuat dengan mengombinasikan sejumlah pengklasifikasi sederhana (lemah).

Persamaan AdaBoost adalah:

$$F(x) = \sum_{t=1}^{T} \propto_t h_t(x) \qquad (3)$$

Di mana:

$h_t(x)$: Pengklasifikasi dasar atau lemah

$\propto_t$: Tingkat pembelajaran (*learning rate*)

F(x) : Hasil, berupa pengklasifikasi kuat atau akhir

Algoritma AdaBoost (Zhou & Yu, AdaBoost, 2009, p. 130):

Masukan:

Dataset D = $\{(x_1,y_1), (x_2,y_2), ..., (x_m,y_m)\}$;

Algoritma pembelajaran lemah (*Weak Learner*) L;

Sebuah *integer* T yang menspesifikasi banyaknya iterasi.

Proses:

Inisialisasi berat distribusi: $D_1(i) = \frac{1}{m}$ untuk semua $i = 1, \dots, m$

for t=1, ... , T:

Melatih pembelajar dasar/lemah h_t dari D menggunakan distribusi D_t

$h_t = L(D,D_t)$

Mengkalkulasi kesalahan dari h_t: $\varepsilon_t = Pr_{x\sim D_t,y} I[h_t(x_i) \neq y_i]$

if $\varepsilon_t > 0.5$ then break

Menetapkan berat dari h_t: $\propto_t = \frac{1}{2}\ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$

$$\frac{D_t(i)}{Z_t} \text{x} \begin{cases} exp(-\propto_t) \; if \; h_t(x_i) = y_i \\ exp(\propto_t) \; if \; h_t(i) \neq y_i \end{cases} \quad (4)$$

Meng-*update* distribusi, di mana Zt adalah faktor normalisasi yang mengaktifkan D_{t+1} menjadi distribusi:

$$\frac{D_t(i)exp(-\propto_t y_i h_t(x_i))}{Z_t} \quad (5)$$

end

Keluaran:

Pengklasifikasi akhir/kuat:

$$H(x) = sign(\sum_{t=1}^{T} \propto_t h_t(x)) \quad (6)$$

Algoritma Naïve Bayes

Pendekatan Bayes penuh mengandung banyak masalah terkait dengan estimasi, sehingga diusulkan Naïve Bayes dengan asumsi bahwa semua atribut bersifat independen (bebas), hal ini menyebabkan kesederhanaan, tetapi sangat efektif dalam praktik (Zaki & Jr, 2014, p. 524). Naïve Bayes adalah salah satu algoritma klasifikasi yang paling efektif dan efisien (Zhang, Jiang, & Su, 2005, p. 1020). Naïve Bayes memiliki keuntungan yang signifikan karena sebagian besar alternatifnya sangat sederhana (Korb & Nicholson, 2011, p. 206).

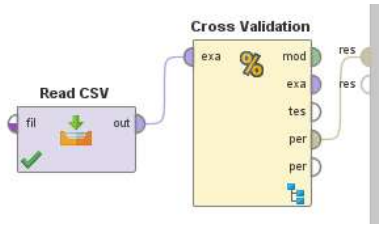
Berikut ini adalah algoritma Naïve Bayes:

a. Masukan: Data latih T, data uji x
b. Hitung *mean* (rata-rata) dan standar deviasi setiap kelas
c. Hitung nilai probabilitas data uji untuk setiap kelas
d. Klasifikasikan data uji sesuai nilai probabilitas kelas yang tertinggi
e. Keluaran: Hasil klasifikasi

4. HASIL EKSPERIMEN

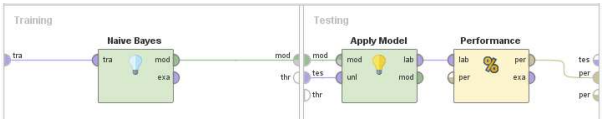
Model Naïve Bayes

Dataset yang disimpan dalam format CSV (*Comma Separated Value*) dibuka menggunakan operator Read CSV. Kemudian keluarannya dihubungkan ke *Cross Validation* (*X-Validation*) dengan nilai *number of folds* 10, karena menerapkan *10-fold cross validation*. Susunan validasi model Naïve Bayes ditunjukkan pada Gambar 4.



Gambar 4 Susunan Validasi Model Naïve Bayes

Untuk melatih dan menguji model di bagian *training* diisi operator Naïve Bayes, pada *testing* diisi operator *Apply Model* dan *Performance*. Susunan operator *training* dan *testing* model Naïve Bayes ditunjukkan pada Gambar 5.



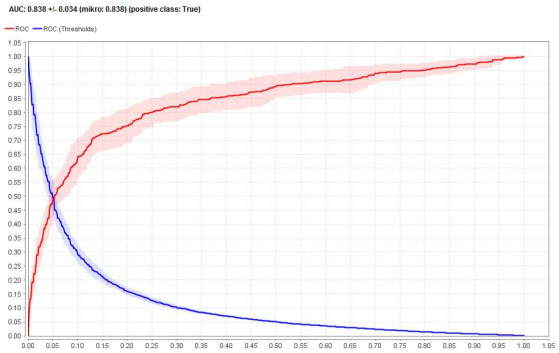
Gambar 5 Susunan Operator Model Naïve Bayes

Kemudian model Naïve Bayes yang telah disusun dieksekusi sehingga didapat hasil kinerja model seperti pada Gambar 6 dan Gambar 7 didapat akurasinya 88,51% dan AUC 0,838.

accuracy: 88.51% +/- 0.75% (mikro: 88.51%)

	true False	true True	class precision
pred. False	2724	257	91.38%
pred. True	126	226	64.20%
class recall	95.58%	46.79%	

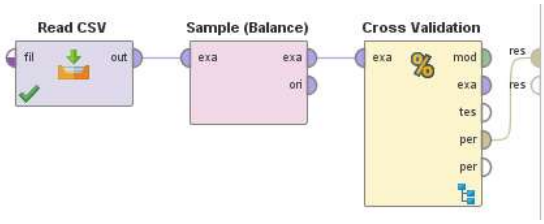
Gambar 6 Akurasi Model Naïve Bayes



Gambar 7 AUC Model Naïve Bayes

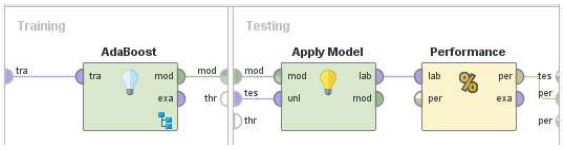
Model Integrasi Random Oversampling, AdaBoost, dan Naïve Bayes

Dataset yang disimpan dalam format CSV (*Comma Separated Value*) dibuka menggunakan operator Read CSV. Untuk menyeimbangkan kelas *churn* dan *nonchurn*, maka digunakan operator *Sample (Balance)*. Untuk menerapkan *random oversampling* (ROS), maka nilai *number of examples per class* diisi 2850 sesuai nilai kelas mayoritas. Kemudian keluarannya dihubungkan ke *Cross Validation* (*X-Validation*) dengan nilai *number of folds* 10, karena menerapkan *10-fold cross validation*. Susunan validasi model integrasi *random oversampling*, AdaBoost, dan Naïve Bayes ditunjukkan pada Gambar 8.

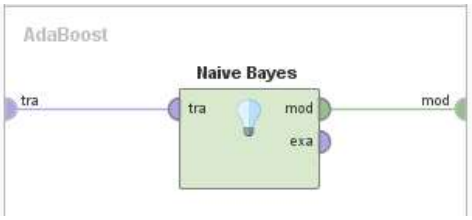


Gambar 8 Susunan Validasi Model Integrasi *Random Oversampling*, AdaBoost, dan Naïve Bayes

Untuk melatih dan menguji model di bagian *training* diisi operator AdaBoost, pada *testing* diisi operator *Apply Model* dan *Performance*. Susunan operator *training* dan *testing* model integrasi *random oversampling*, AdaBoost, dan Naïve Bayes ditunjukkan pada Gambar 9 dan Gambar 10.



Gambar 9 Susunan Operator Model Integrasi *Random Oversampling*, AdaBoost, dan Naïve Bayes



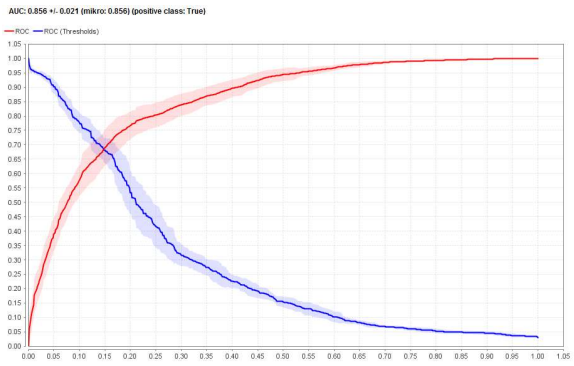
Gambar 10 Susunan Operator AdaBoost

Kemudian model integrasi *random oversampling*, AdaBoost, dan Naïve Bayes yang telah disusun dieksekusi sehingga didapat hasil kinerja model seperti pada Gambar 11 dan Gambar 12 didapat akurasinya 78,30% dan AUC 0,856.

accuracy: 78.30% +/- 2.07% (mikro: 78.30%)

	true False	true True	class precision
pred. False	2255	642	77.84%
pred. True	595	2208	78.77%
class recall	79.12%	77.47%	

Gambar 11 Akurasi Model Integrasi *Random Oversampling*, AdaBoost, dan Naïve Bayes

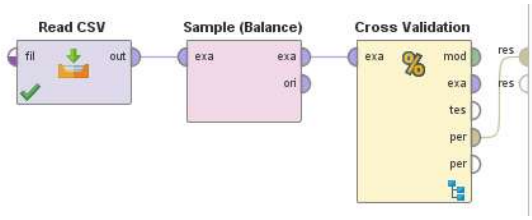


Gambar 12 AUC Model Integrasi *Random Oversampling*, AdaBoost, dan Naïve Bayes

Model Random Undersampling, AdaBoost, dan Naïve Bayes

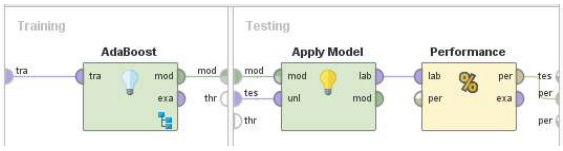
Dataset yang disimpan dalam format CSV (*Comma Separated Value*) dibuka menggunakan operator Read CSV. Untuk menyeimbangkan kelas *churn* dan *nonchurn*, maka digunakan operator *Sample (Balance)*. Untuk menerapkan *random undersampling* (ROS), maka nilai *number of examples per class* diisi 483 sesuai nilai kelas minoritas dan *allow downsampling* diberi tanda centang (*check*). Kemudian keluarannya dihubungkan ke *Cross Validation* (*X-Validation*) dengan nilai *number of folds* 10, karena menerapkan *10-fold cross validation*. Susunan validasi model integrasi *random undersampling*,

AdaBoost, dan Naïve Bayes ditunjukkan pada Gambar 13.

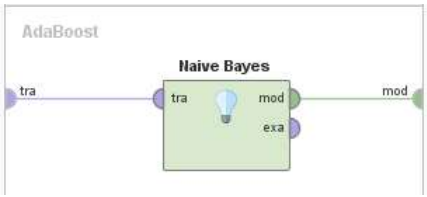


Gambar 13 Susunan Validasi Model Integrasi *Random Undersampling*, AdaBoost, dan Naïve Bayes

Untuk melatih dan menguji model di bagian *training* diisi operator AdaBoost, pada *testing* diisi operator *Apply Model* dan *Performance*. Susunan operator *training* dan *testing* model integrasi *random undersampling*, AdaBoost, dan Naïve Bayes ditunjukkan pada Gambar 14 dan Gambar 15.



Gambar 14 Susunan Operator Model Integrasi *Random Undersampling*, AdaBoost, dan Naïve Bayes



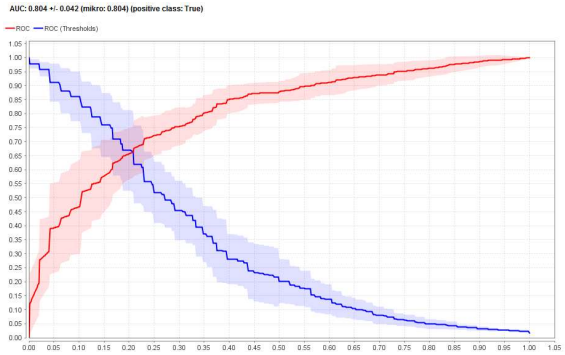
Gambar 15 Susunan Operator AdaBoost

Kemudian model integrasi *random undersampling*, AdaBoost, dan Naïve Bayes yang telah disusun dieksekusi sehingga didapat hasil kinerja model seperti pada Gambar 16 dan Gambar 17 didapat akurasinya 74,33% dan AUC 0,804.

accuracy: 74.33% +/- 4.52% (mikro: 74.33%)

	true False	true True	class precision
pred. False	354	119	74.84%
pred. True	129	364	73.83%
class recall	73.29%	75.36%	

Gambar 16 Akurasi Model Integrasi *Random Undersampling*, AdaBoost, dan Naïve Bayes



Gambar 17 AUC Model Integrasi *Random Undersampling*, AdaBoost, dan Naïve Bayes

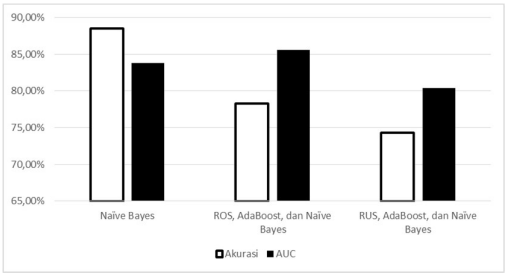
5. PEMBAHASAN

Dari penerapan dan validasi model yang diusulkan diperoleh ukuran akurasi dan AUC seperti ditunjukkan pada Tabel 2.

Tabel 2 Hasil Pengukuran Kinerja Model

Model	Akurasi	AUC
Naïve Bayes	88,51%	0,838
ROS, AdaBoost, dan Naïve Bayes	78,30%	0,856
RUS, AdaBoost, dan Naïve Bayes	74,33%	0,804

Hasil pengukuran kinerja yang didapat divisualisasikan dalam grafik batang 3D seperti pada Gambar 18. Dari grafik tersebut terlihat bahwa akurasi model Naïve Bayes memiliki nilai tertinggi, tetapi nilai AUC tertinggi dimiliki oleh model integrasi ROS, AdaBoost, dan Naïve Bayes. Karena model prediksi kecenderungan pelanggan mengalami churn lebih mengutamakan akurasi dalam menemukan kelas churn, maka model terbaik adalah model yang memiliki nilai AUC tertinggi, yaitu model integrasi *random oversampling*, AdaBoost, dan Naïve Bayes.



Gambar 18 Visualisasi Kinerja Model

6. KESIMPULAN

Model integrasi *random oversampling*, AdaBoost, dan Naïve Bayes memiliki akurasi lebih rendah, tetapi memiliki nilai AUC lebih tinggi. Hal

ini menunjukkan bahwa model integrasi *random oversampling*, AdaBoost, dan Naïve Bayes memiliki kinerja lebih baik dalam menemukan kecenderungan pelanggan yang mengalami *churn*, sedangkan Model integrasi *random undersampling*, AdaBoost, dan Naïve Bayes memiliki akurasi dan nilai AUC yang lebih rendah, sehingga tidak tepat jika digunakan untuk prediksi *churn* pelanggan.

DAFTAR PUSTAKA

Afza, A. J., Farid, D. M., & Rahman, C. M. (2011). A Hybrid Classifier using Boosting, Clustering, and Naïve Bayesian Classifier. *World of Computer Science and Information Technology Journal (WCSIT)*, 105-109.

Catal, C. (2012). Performance Evaluation Metrics for Software Fault Prediction Studies. *Acta Polytechnica Hungarica, 9*(4), 193-206.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, 321–357.

Chen, Z.-Y., Fan, Z.-P., & Sun, M. (2012). A Hierarchical Multiple Kernel Support Vector Machine for Customer Churn Prediction Using Longitudinal Behavioral Data. *European Journal of Operational Research, 223*(2), 461-472. doi:10.1016/j.ejor.2012.06.040

Churi, A., Divekar, M., Dashpute, S., & Kamble, P. (2015). Analysis of Customer Churn in Mobile Industry using Data Mining. *International Journal of Emerging Technology and Advanced Engineering, 5*(3), 225-230. Retrieved from www.ijetae.com/files/Volume5Issue3/IJETAE_0315_41.pdf

Harrington, P. (2012). *Machine Learning in Action.* New York: Manning Publications Co.

Huang, B., Kechadi, M. T., & Buckley, B. (2012). Customer Churn Prediction in Telecommunications. *Expert Systems with Applications*, 1414-1425.

Jadhav, R. J., & Pawar, U. T. (2011). Churn Prediction in Telecommunication Using Data Mining Technology. *(IJACSA) International Journal of Advanced Computer Science and Applications, 2*(2), 17-19.

Keramati, A., Jafari-Marandi, R., Aliannejadi, M., Ahmadian, I., Mozzafari, M., & Abbasi, U. (2014). Improved Churn Prediction in Telecommunication Industry Using Data Mining Techniques. *Applied Soft Computing, 24*, 994–1012. doi:10.1016/j.asoc.2014.08.041

Korada, N. K., Kumar, N. P., & Deekshitulu, Y. (2012). Implementation of Naïve Bayesian Classifier and Ada-Boost Algorithm Using Maize Expert System. *International Journal of Information Sciences and Techniques (IJIST) Vol.2, No.3*, 63-75.

Korb, K. B., & Nicholson, A. E. (2011). *Bayesian Artificial Intelligence* (2nd ed.). Florida: CRC Press.

Lu, J. (2002). Predicting Customer Churn in the Telecommunications Industry - An Application of Survival Analysis Modeling Using SAS. *Proceedings of the Twenty-Seventh Annual SAS® Users Group International Conference* (pp. 1-6). Orlando: SAS Institute Inc. Retrieved from http://www2.sas.com/proceedings/sugi27/p114-27.pdf

Nistanto, R. K. (2014, Juni 4). *Tekno: 2015, Pengguna "Mobile" Lampaui Jumlah Penduduk Dunia*. Retrieved from kompas.com: http://tekno.kompas.com/read/2014/06/04/1025003/2015.pengguna.mobile.lampaui.jumlah.penduduk.dunia

Peng, Y., & Yao, J. (2010). AdaOUBoost: Adaptive Over-sampling and Under-sampling to Boost the Concept Learning in Large Scale Imbalanced Data Sets. *Proceedings of the international conference on Multimedia information retrieval* (pp. 111-118). Philadelphia, Pennsylvania, USA: ACM.

Rodan, A., Fayyoumi, A., Faris, H., Alsakran, J., & Al-Kadi, O. (2015). Negative Correlation Learning for Customer Churn Prediction: A Comparison Study. *The Scientific World Journal*, 1-7. doi:10.1155/2015/473283

Sanou, B. (2015). *ICT Facts & Figures.* Geneva: International Telecommunication Union.

Sun, Y., Mohamed, K. S., Wong, A. K., & Wang, Y. (2007). Cost-sensitive Boosting for Classification of Imbalanced Data. *Pattern Recognition Society*, 3358-3378.

Umayaparvathi, V., & Iyakutti, K. (2012). Applications of Data Mining Techniques in Telecom Churn Prediction. *International Journal of Computer Applications, 42*(20), 5-9. doi:10.5120/5814-8122

Verbeke, W., Dejaeger, K., Martens, D., Hur, J., & Baesens, B. (2012). New Insights into Churn Prediction in the Telecommunication Sector: A Profit Driven Data Mining Approach. *European Journal of Operational Research, 218*(1), 211-229. doi:10.1016/j.ejor.2011.09.031

Witten, I. H., Frank, E., & Hall, M. A. (2011). *Data Mining: Practical Machine Learning Tools and Techniques* (3rd ed.). Burlington: Morgan Kaufmann.

Yap, B. W., Rani, K. A., Rahman, H. A., Fong, S., Khairudin, Z., & Abdullah, N. N. (2014). An Application of Oversampling, Undersampling, Bagging and Boosting in Handling Imbalanced Datasets. *Proceedings of the First International Conference on Advanced Data and Information Engineering (DaEng-2013). 285*, pp. 13-22. Singapore: Springer. doi:10.1007/978-981-4585-18-7_2

Yu, X., Guo, S., Guo, J., & Huang, X. (2010). An Extended Support Vector Machine Forecasting Framework for Customer Churn in E-Commerce. *Expert Systems with Applications, 38*(3), 1425-1430. doi:10.1016/j.eswa.2010.07.049

Zaki, M. J., & Jr, W. M. (2014). *Data Mining and Analysis: Fundamental Concepts and Algorithms.* New York: Cambridge University Press.

Zhang, D., Liu, W., Gong, X., & Jin, H. (2011). A Novel Improved SMOTE Resampling Algorithm Based on Fractal. *Computational Information Systems*, 2204-2211.

Zhang, H., Jiang, L., & Su, J. (2005). Augmenting Naïve Bayes for Ranking. *ICML '05 Proceedings of the 22nd international conference on Machine learning* (pp. 1020 - 1027). New York: ACM Press. doi:http://dx.doi.org/10.1145/1102351.1102480

Zhou, Z.-H., & Yu, Y. (2009). *The Top Ten Algorithms in Data Mining.* (X. Wu, & V. Kumar, Eds.) Florida: Chapman & Hall/CRC.